Basic Concepts in Augmented Reality Audio

Jacques Lemordant

W3C workshop: Augmented Reality on the web

Position Paper

Title: Basic Concepts in Augmented Reality Audio

Jacques lemordant
jacques.lemordant@inria.fr
http://wam.inrialpes.fr/people/lemordant/

1. Introduction

An AR system may be defined as a system having the three following characteristics:

- combines real world and virtual objects
- is interactive or reactive
- uses 3D positioning of virtual objects

This definition is clearly applicable to Augmented Reality Audio (ARA) systems as we shall see.

2. Real, virtual, and augmented audio environments

The basic difference between real and virtual sound environments is that virtual sounds are originating from another environment or are artificially created, whereas the real sounds are the natural existing sounds in the user's own environment. Augmented Reality Audio combines these aspects in a way where real and virtual sound scenes are mixed so that virtual sounds are perceived as an extension or a complement to the natural ones.

Augmented Reality Audio is used in many mobile applications like geolocalized games, non-linear audio walking tours, navigation systems for visually impaired people. Different types of navigation will require different types of applications. For example, a mountain biker navigation application will be very different from a guidance application for visually impaired people.

The rendering of an ARA scene can be experimented through the use of bone conduction headsets, headphones with integrated microphones or earphones with acoustically transparent earpieces, with the audio being played by a mobile phone. ARA applications can be designed so that they do not interfere with the user practicing other activies, i.e., the application leaves the user's hands free and does not require visual attention from the user.

All of the three characteristics of an AR system set different requirements for ARA software and hardware.

- an ARA scene has to be authored through the join use of two XML languages, one for the representation of the real world and the other one for the representation of the 3D virtual audio scene, the link between the two being done through a tag-based dispatching language.
- for interaction in mobile usage, outdoor tracking has to be done through GPS and indoor tracking through embedded sensors like accelerometers and magnetometers or external ranging sensors. A user of an ARA application can interact via the microphones in the headset, speech or sound recognition being be used for controlling the application. The audio language must be an event-based language to allow interactive audio through instantiation of sound models.

- for 3D rendering of sound objects, the user's position and orientation need to be estimated in real time. With real-time head tracking the virtual sound objects can be tied to the surrounding environment and thus the virtual environment stays in place even if the user moves. Another benefit with real-time head tracking is that the dynamic cues from head rotation help localizing sound objects in the environment, especially for front-back confusion.

In our position is that:

1. Audio is an essential part of an Augmented Reality System especially in the mobile case
2. An XML Format for Interactive Audio and its associated DOM/event API have to be used to describe the 3D virtual audio scene.

3. Interactive Audio

I*nteractive audio is here to stay and will continually increase its presence in all types of human experiences. It makes the greatest sense to get the foundation properly established at the outset to avoid scattered efforts and incompatibilities. That is why it is important to start now to develop a formal understanding of the main principles of what constitutes interactive audio so we can collectively design and share customized instances of those building blocks and simplify the design and production process for all industries seeking to enhance their products with interactive audio* [1]

3.1 Reactive vs. Interactive

Not all systems that respond to input stimuli can be defined as interactive audio systems. An interactive audio system allows changes in input behavior to modify the audio behavior, whereas a reactive system simply plays back static audio events without any adaptation to the user stimulus. An ARA guidance application cannot be only reactive and the behavior of sound objects has to be modified accordinly to the context.

3.2 Direct vs. Indirect Input Stimuli

The input stimuli to the system can be classified into two categories. In the direct case, the user is consciously controlling the audio; in the indirect case, the user is controlling some other parameter that in turn affects the audio. For example a video game player indirectly interacts with the audio like the user of a guidance application.

To design an XML language for Interactive Audio, it is important to develop a formal understanding of the main principles of what constitutes interactive audio. This XML language will allow to collectively design and share customized instances of an ARA system and simplify the design and production process for all industries seeking to enhance their products with interactive audio [1].

4. Sound Objects or sound structuration

In ARA the main interface for giving information to the user is the audio system. This kind of an audio-only way of conveying information is called Auditory Display. The auditory display can be spread around the user in 3D and the information given as recorded or virtual speech, non-speech sounds, such as earcons or auditory icons, or a combination of all of these. Earcons are structured sequences of sounds that can be used in different

combinations to create complex audio messages, whereas auditory icons are everyday sounds used to convey information to the user. We can go further with the concept of sound objects.

Initially, a sound object as defined by Pierre Schaeffer[4] is a generalization of the concept of a musical note, i.e, any sound from any source which in duration is on the time scale of 100 ms to several seconds. This concept can be transposed and extended through a hierarchical structuring of sounds with internal/external synchronisation and DSP parametrization.

The concept of sound objects allows for:
- Better organization (sound classification)
- Easy non-linear audio cues creation / randomization
- Better memory usage by the use of small audio chunks (common parts of audio phrases can be shared)
- Allows separate mixing of cues to deal with priority constraints easily
- Reusability

In iXMF[1], sound objects are called cues and we have followed this terminology in A2ML which is a SMIL-based version of iXMF.
The A2ML fragment below (see [2] for a more detailed document) contains cues models to be instantiated by events:

```
<cue id="ambiance" loopCount="-1" begin="environment.ambiance">
    <chunk pick="fixed">
        <sound src="/environment/
ambiance_office.wav" setActive="environment.set_ambiance.office"/>
        <sound src="/environment/
ambiance_hall.wav" setActive="environment.set_ambiance.hall"/>
    </chunk>
</cue>

<cue id="floor_surface" loopCount="1" begin="environment.floor_surface_change">
   <chunk pick="fixed">
      <sound src="/environment/floor_surface_carpet.wav"
 setActive="environment.set_floor_surface.carpet"/> src="/environment/
floor_surface_marble.wav" setActive="environment.set_floor_surface.marble"/>>
      </sound>
   </chunk>
</cue>

<cue id="way" loopCount="1" begin="environment.way">
   <chunk>
      <sound src="/environment/way.wav"/>
   </chunk>
</cue>
```

5. Mixing groups or style specification

Like in a traditional mixing console, mix groups can be used to regroup multiple cues and apply mix parameters on all of them at the same time. In our format, we called them *sections* as, in addition to mixing multiple cues, they can also be used to add DSP effects and locate the audio in a virtual 3D environment. The main difference with traditional mix

groups is that a cue can be a member of multiple sections, and the effects of all of them will apply, making sections very versatile.The sound manager's response to a given cue instantiation may be simple, such as playing or halting a 3D sound source, or it may be complex, such as dynamically manipulating various DSP parameters over time. The sound manager is also offering an lower level API through which all instance parameters can be manipulated such as positions of the sound sources and the auditor.

Below is an example of rendering or style specification in A2ML.
As can be seen SMIL animation of parameters is supported.

```
<sections>
   <!-- Mix group for the global audio guide. Use the reverb as a way to notify room size
changes. -->
   <section id="audioguide" cues="next_wp door stairway elevator elevator_button
ambiance floor_surface way">
         <dspControl dspName="reverb">
               <parameter name="preset" value="default"/>
               <animate id="preset_change"
 attribute="preset" values="env.change_reverb_preset"/>
         </dspControl>
         <volumeControl level="70"/>
   </section>

    <!-- Activates 3D positioning for the object that need it. Position of the objects is
controlled by the guidance application. -->
   <section id="objects3D" cues="next_wp door stairway elevator floor_surface">
      <mix3D> <distanceAttenuationControl attenuation="2
      </mix3D>
    </section>

    <!-- Submix group for the environment details. -->
    <section id="details" cues="atrium_door_number">
       <mix3D>
         <distanceAttenuationControl attenuation="5 </mix3D>
         <volumeControl level="100"/>
   </section>
</sections>
```

References

[1] Project barbq, http://www.projectbarbq.com/bbq03/bbq03r5.htm.
[2] Augmented Reality Audio Editing. Jacques lemordant, Yohan Lasorsa, *128th AES Convention*, 2010, http://wam.inrialpes.fr/publications/index.en.html.
[3] An Interactive Audio System for Mobiles. Yohan Lasorsa, Jacques lemordant, *127th AES Convention*, 2009. http://wam.inrialpes.fr/publications/index.en.html.
[4] Sound object, Pierre Schaeffer 1959, http://en.wikipedia.org/wiki/Sound_object